Communications of the Association for Information Systems

Volume 44 Article 32

5-2019

An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem

Dominik Bork
University of Vienna, dominik.bork@univie.ac.at

Robert Andrei Buchmann
Babes-Bolyai University

Dimitri Karagiannis
University of Vienna

Moonkun Lee
Chonbuk National University

Elena-Teodora Miron
OMiLAB NPO

Recommended Citation

Bork, D., Buchmann, R. A., Karagiannis, D., Lee, M., & Miron, E. (2019). An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem. Communications of the Association for Information Systems, 44, pp-pp. https://doi.org/10.17705/1CAIS.04432

Communications of the Association for Information Systems

Research Paper DOI: 10.17705/1CAIS.04432 ISSN: 1529-3181

An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem

Dominik Bork
Research Group Knowledge Engineering
University of Vienna
Austria
dominik.bork@univie.ac.at

Robert Andrei Buchmann
Business Informatics Research Center
Babes-Bolyai University
Romania

Dimitris Karagiannis
Research Group Knowledge Engineering
University of Vienna,
Austria

Moonkun Lee
Chonbuk National University
Republic of Korea

Elena-Teodora Miron
OMiLAB NPO
Germany

Abstract:

This paper motivates, describes, demonstrates in use, and evaluates the Open Models Laboratory (OMiLAB)—an open digital ecosystem designed to help one conceptualize and operationalize conceptual modeling methods. The OMiLAB ecosystem, which a generalized understanding of "model value" motivates, targets research and education stakeholders who fulfill various roles in a modeling method's lifecycle. While we have many reports on novel modeling methods and tools for various domains, we lack knowledge on conceptualizing such methods via a full-fledged dedicated open ecosystem and a methodology that facilitates entry points for novices and an open innovation space for experienced stakeholders. This gap continues due to the lack of an open process and platform for 1) conducting research in the field of modeling method design, 2) developing agile modeling tools and model-driven digital products, and 3) experimenting with and disseminating such methods and related prototypes. OMiLAB incorporates principles, practices, procedures, tools, and services required to address the issues above since it focuses on being the operational deployment for a conceptualization and operationalization process built on several pillars: 1) a granularly defined "modeling method" concept whose building blocks one can customize for the domain of choice, 2) an "agile modeling method engineering" framework that helps one quickly prototype modeling tools, 3) a model-aware "digital product design lab", and 4) dissemination channels for reaching a global community. In this paper, we demonstrate and evaluate the OMiLAB in research with two selected application cases for domain- and case-specific requirements. Besides these exemplary cases, OMiLAB has proven to effectively satisfy requirements that almost 50 modeling methods raise and, thus, to support researchers in designing novel modeling methods, developing tools, and disseminating outcomes. We also measured OMiLAB's educational impact.

Keywords: OMiLAB, Open Models, Domain-specific Conceptual Modeling, Model Value, Agile Modeling Method Engineering.

This manuscript underwent peer review. It was received 12/23/2018 and was with the authors for 6 months for 2 revisions. Tom Eikebrokk served as Associate Editor.

1 Introduction

In the advent of digital transformation and emerging paradigms such as the Internet of things or Industry 4.0, one needs to design innovative product-service systems to support changes in the way humans employ information technology and engage in processes. These changes come with novel requirements for modeling methods and tools. Modeling standards such as Business Process Model and Notation (BPMN) and Unified Modeling Language (UML) have a limited ability to cope with the needs of such emerging paradigms for three reasons. First, they have general applicability and, thus, focus a higher level of abstraction—they ignore domain-specific aspects to a large extent. Second, they adopt broad adoption and stability as their core value; therefore, they feature rather long update cycles and incremental updates. Third, they hardly operationalize the knowledge that the conceptual models codify. Accordingly, recent business information systems engineering research has explored novel domain-specific conceptual modeling languages (Frank et al., 2014) and developed modeling tools that enable their efficient application (Brenner et al., 2014). Conceptual modeling has become an established research field in information systems (see Recker, 2015). Simultaneously, an increasing amount of interest has examined openness in business and information systems engineering research (van der Aalst, Bichler, & Heinzl, 2016).

The proposal we present in this paper originates from a generalized notion of “model value” and requirements with respect to research and teaching that arise from it. We introduce the Open Models Laboratory (OMiLAB) as a digital ecosystem that fosters open innovation in conceptual modeling research and teaching. OMiLAB has accumulated results during several years of operation. At its implementation core, it constitutes various enablers that together deploy a conceptualization and operationalization process for modeling methods. Strategically, it is an open community and resource repository that supports “modeling for the masses”, which the European enterprise modeling community recently made explicit and roadmapped (Sandkuhl et. al., 2018). By generalizing the idea of model value beyond traditionally established application areas (business process management, software engineering, etc.), it also reflects the value creation principles in the *Memorandum on Design-oriented Information Systems Research* (Österle et al. 2011). In other words, it treats models and modeling methods not only as fixed artifacts observed through a behavioristic lens but also as dynamic artifacts subjected to design-related concerns and agility requirements that benefit from accumulated community-sourced practices, design patterns, and lessons learned.

OMiLAB addresses three grand research challenges that Becker, vom Brocke, Heddier, and Seidel (2015, p. 382) raise: 1) “supporting effective collaboration and learning through evolving media repertoires”, “raising collective consciousness”, and “developing model-driven methods and tools for the full-scale automated generation of implementation-ready IS”. At the same time, the OMiLAB recognizes pitfalls and fallacies with respect to implementing open innovation communities (von Briel & Recker, 2017). Following these challenges, the OMiLAB focuses on establishing an ecosystem platform that enables prototyping, experimentation, and practice-oriented information science research (e.g., related to method engineering) (Bucher, Klesse, Kurpjuweit, & Winter, 2007). It also focuses on acting as an open community facilitator for conceptual modeling similar to how Wikipedia supports general purpose knowledge creation and distribution or how platforms such as Apple iOS and Google Android established mobile apps ecosystems (see Benlian, Hilkert, & Hess, 2015). The community shares experience and knowledge assets, raises modeling requirements, and evaluates prototypical solutions.

This vision becomes manifest in the pillars the OMiLAB builds on: 1) a granularly defined “modeling method” concept whose building blocks can be customized for specific domains, 2) an “agile modeling method engineering” (AMME) framework that helps one quickly prototype modeling tools, 3) a model-aware “digital product design lab”, and 4) dissemination channels for reaching a global community. These pillars enable Sabegh and Recker’s (2017b) new conceptual modeling research framework, which pushes forward the conceptual modeling agenda along several directions (e.g.,) novel forms of representation, agile modeling, modeling for other purposes than information systems development).

This paper proceeds as follows: in Section 2, we outline the problem and the derived requirements and summarize how the OMiLAB addresses them. In Section 3, we detail the conceptualization process that the OMiLAB digital ecosystem enables. In Section 4, we describe the OMiLAB’s open innovation community. In Section 5, we report on two modeling method conceptualization cases. We compare related approaches to the OMiLAB in Section 6. In Section 7, we evaluate the OMiLAB via multiple facets. In Section 8, we discuss future research directions and conclude the paper.

2 Problem Statement and Solution Overview

In information systems (IS) education, educators often teach conceptual modeling as subsidiary to other fields (e.g., business process management, database design, software engineering) and rely on well-established modeling languages that satisfy some fundamental use cases (process simulation, code generation, etc.). As a result, they perceive models as having limited value (i.e., they do not perceive conceptual modeling as a discipline in and of itself) (Buchmann & Ghiran, 2017). Instead, they fragment conceptual modeling into several techniques attached to other fields and view it as subordinate to the modeling languages they select. In research, on the other hand, conceptual modeling challenges often extend well beyond those teaching cases and often consider diverse paradigms: knowledge management, enterprise architecture management, design science, and so on. In a design science context (Hevner, March, Park, & Ram, 2004), one may consider a modeling tool, language, or method a standalone artifact that does not have strictly instrumental value as a means to some end that software engineering defines.

Consequently, a gap between the two perspectives becomes inherent. In order to bridge it, researchers and practitioners must have the ability to conceptualize modeling methods that are themselves artifacts and that they value with respect to requirements emerging from domain-specific or even case-specific needs. As such, modeling stakeholders require a **conceptualization process** to support this vision and an **ecosystem** where they can easily become productive in **operationalizing** it. In this paper, we introduce the OMiLAB, an exemplar of such an ecosystem built around a conceptualization and operationalization process for attaining a generalized notion of “model value”.

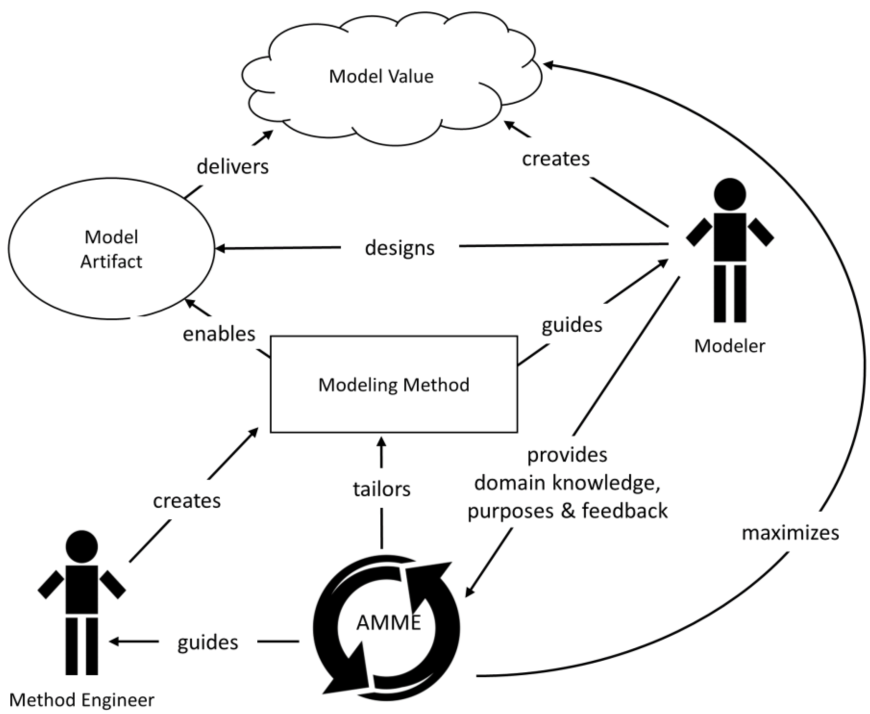

Figure 1. The Key OMiLAB Motivator: The Co-creation of Model Value

Model value must be detached from a particular usage or modeling language. It needs to encompass both *instrumental value* (as enabler for a model-aware digital product) and *intrinsic value* (as a means of knowledge representation). At the same time, it must align with contemporary value theories such as “value co-creation”, an economic strategy of stimulating collaboration for mutually valued outcome (Galvagno & Dalli, 2014). However, co-creation implies interactive relationships and at least two contributing roles, which Figure 1 illustrates in the OMiLAB context:

1) The *modeler*: as the main source of modeling requirements and the individual who designs model artifacts, the modeler has a direct influence on model value. This value is linked to purposes (reflected in modeling requirements) that may evolve (reclaiming agility), and is largely determined by the knowledge that can be acquired while engineering the modeling

method. Furthermore, the modeler also acts as the primary source of feedback, thereby steering the evolution and improvement of the modeling method;

2) The *method engineer:* guided by the agile modeling method engineering framework (Karagiannis, 2015), the method engineer customizes a modeling method according to evolving needs and a gradual understanding of the application domain and, thus, enables model value and maximizes it while fine tuning the model's expressiveness and providing adequate guidance to the modeler.

In this paper, we explore the following research question:

RQ: How can we enable modeling stakeholders to co-create model value in an agile manner while considering both instrumental and intrinsic criteria?

As a solution, we propose a conceptualization process and an ecosystem to operationalize this process and the resulting model-aware digital product. The OMiLAB operationally deploys the conceptualization process that we propose to support the co-creation strategy in Figure 1, which, eventually, may become the core principle of an economy of model value chains. In the long term, this digital ecosystem seeks to observe such value chains and refine co-creation models across various projects.

We abstracted our research question from the requirements of more than 50 projects on the OMiLAB. We discuss these requirements on a more granular level in Table 1 and summarize how various artifacts that the OMiLAB provides address them in subsequent sections.

To address these requirements, we established the conceptualization process underlying the OMiLAB digital ecosystem with its key community roles and building blocks as we visualize in Figure 2. Depending on project complexity and requirements, the same actor may fulfill multiple stakeholder roles or multiple contributors may share the same role:

- *Domain expert*: the source of domain knowledge that must be acquired and embedded in all building blocks of a modeling method (i.e., in notational specificity (see Bork, Karagiannis, & Pittl, 2018a), in concept definitions, in functionality reflecting some aspects of business logic).
- *Metamodel designer*: responsible for the central building block of a modeling method (i.e., the modeling language definition, which covers notation, syntax, and semantics).
- *Modeling method engineer*: builds the modeling method by extending the language with a modeling procedure and model-driven functionality and implements them in a usable modeling tool to achieve qualities such as adaptability, extensibility, usability, operability and integrability. The modeling method engineer employs the AMME framework (Karagiannis, 2015) to achieve this output.
- *Digital product engineer*: applies the modeling method for a selected domain or case in order to realize a digital product that employs the modeling tool as its “knowledge engine” (i.e., a knowledge-acquisition enabler that interoperates with other artifacts (e.g., Internet of things components, microservices) either through direct interoperability mechanisms or through the model base). Thus, digital product engineers can build or experiment on digital product prototypes that employ models as a knowledge source. An m:n relationship exists between the modeling language defined in the first stage and the digital products (e.g., a hybridization of modeling languages may be necessary in the same modeling tool—something that AMME makes possible).
- *The end user*: experiences the digital product prototype, which typically includes a modeling tool and software artifacts that interoperate with the modeling environment through some available channels (e.g., Web service, model serializations). In the absence of a digital product, the end user acts strictly as a modeler in using the modeling tool and its customized functionality (e.g., simulation, reporting, model queries).

Table 1. OMiLAB Requirements and their Respective Treatments

Requirement	How OMiLAB addresses the requirement
The decomposition requirement: need to manage the complexity in modeling methods by separating concerns regarding their building blocks. This need closely relates to agility because different teams may have to develop a modeling method's different components, so components must be allowed to evolve separately (considering possible dependencies).	The OMiLAB conceptualization process was defined around the "seed concept" of modeling method as introduced in (Karagiannis & Kühn, 2002), where its building blocks and dependencies are prescribed in a decompositional manner. Fill, Redmond, and Karagiannis (2012) relate this seed concept to features of the metamodeling platform ADOxx (ADOxx.org, 2018) to help stakeholders formally design and analyze such methods. In discussing their experience from one project, Buchmann and Karagiannis (2015) present different ways to propagate requirements among the modeling method's building blocks.
The agility requirement: need to enable agility in all a modeling method's building blocks (treated as backlog items). We understand agility here as responsiveness to changing needs. Software engineering methods recognize this quality, though modeling methods less commonly feature it. That is, modeling languages are often perceived as invariants with respect to development processes or as striving for universal consensus and standardization.	OMiLAB empowers its stakeholders to conceptualize and rapidly prototype modeling methods based on their application case requirements or to propagate model-aware digital products. An incremental and iterative framework called agile modeling method engineering enables this agility (Karagiannis, 2015). Instead of focusing on researching interoperability mechanisms across existing modeling tools, stakeholders can tailor their own modeling tool either from scratch or by reusing already implemented building blocks in an agile manner—see the BEE-UP open tool (Karagiannis, Buchmann, Burzynski, Reimer, & Walch, 2016b).
The knowledge ecosystem requirement: co-creation raises a need for a shared repository of knowledge assets, practices, and deliverables that can support knowledge acquisition and knowledge transfer among projects, engineering phases, stakeholder types, even application domains.	The OMiLAB provides a portal organized around projects of specific granularity. The OMiLAB considers the prescribed building blocks and the typical deliverables that stakeholders produce when conceptualizing a modeling method, which includes specification documents or evaluation protocols. It also provides the user-management mechanisms to enable a social dimension for interactions in its community.
The openness requirement: co-creation implies interactive relationships, which one can typically find at the core of open innovation communities. Open use and open source implementations and reusable services and tutorials facilitate interaction by lowering entry barriers for novices. At the same time, they provide best practices or reference implementations to domain experts.	Openness represents the fundamental motivator and philosophy for the OMiLAB (Bork & Miron, 2017) with respect to both the technological resources that stakeholders use during the conceptualization and to the knowledge artifacts they create.
The technology requirement: a need for technological enablers that facilitate fast prototyping, considering both the intrinsic value of models (as knowledge assets) and their instrumental value (as input for some model-aware artifacts).	The OMiLAB provides freely usable metamodeling enablers, modeling tools, and various plug-ins for developing model-aware digital products (e.g., model serialization plug-ins, microservice frameworks for model querying/publishing). Further, researchers have begun to develop a platform-independent declarative language for defining a modeling method (Visic, Fill, Buchmann, & Karagiannis, 2015), which suggests that other metamodeling platforms may also become OMiLAB resources as long as their creators want to and can support the foundational philosophy of openness, flexibility, sustainability, and maturity that the OMiLAB advocates.
The dissemination requirement: in academic and scientific communities, researchers have a particular kind of incentive to disseminate their work. They need to publish artifacts or experimental results generated when conceptualizing and operationalizing modeling methods.	The OMiLAB portal and specific services function as a dissemination channel where method engineers manage their own workspace whose structure focuses on the typical artifacts that these engineers build when conceptualizing modeling methods. Additionally, OMiLAB events and joint publications foster a community of innovation and shared understanding of model value.

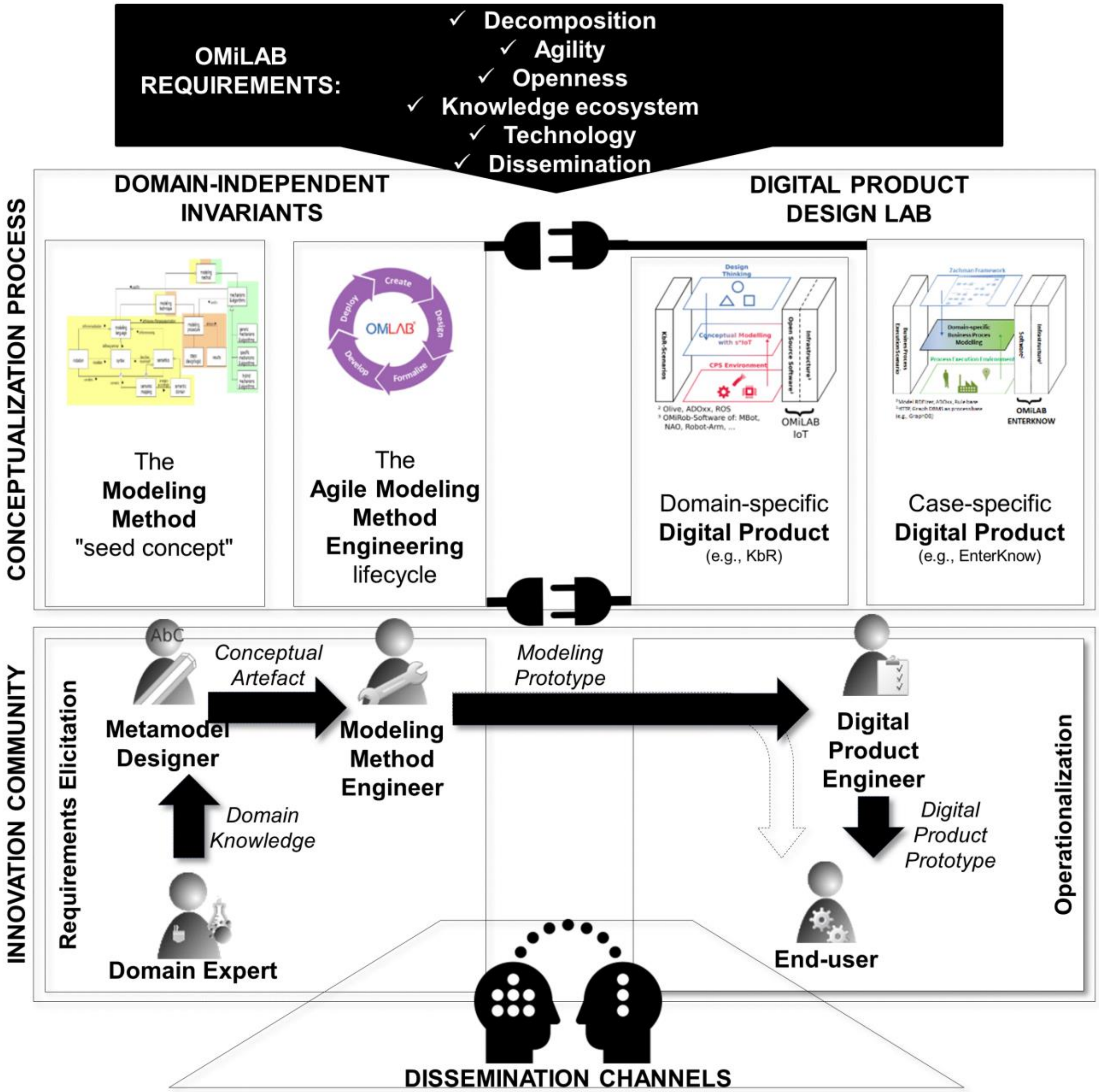

Figure 2. The OMiLAB Digital Ecosystem

Specific enablers (which we discuss in subsequent sections) support all stakeholders in the OMiLAB digital ecosystem (see Figure 2). Each step in the conceptualization process produces artifacts that have a deliverable value (can be disseminated, extended, or adapted for other projects). Figure 3, which includes key OMiLAB resources on the left side and disseminable assets in the right side, provides a process-centric view of the OMiLAB digital ecosystem.

We discuss the environments that form OMiLAB and their respective key resources in Sections 3 and 4. We then summarize two exemplary application cases in Section 5 to illustrate the two most common types of projects that the OMiLAB hosts: 1) one where an evolving modeling tool prototype represents the main outcome (i*; see Section 5.1) and 2) one where the knowledge core of some "digital product proof-of-concept" represents the modeling tool (in our selected case, a business process management system called EnterKnow; see Section 5.2). The second type of project involves domain-specific run-time environments for which interoperability enablers are of essence; also, from a methodological standpoint, they rely on a flavor of model-driven software development that we call "model-aware development" here. Its key characteristic is that the developed software uses a model base at run-time as a complement to its more traditional database/data sources. Model queries (Buchmann et al., 2018) and service-based architectures (Walch & Karagiannis, 2017) are commonly used to implement such model-awareness. From an operationalization standpoint, both types of projects generate artifacts that community members can exploit to achieve their (educational, practical, experimental, etc.) goals.

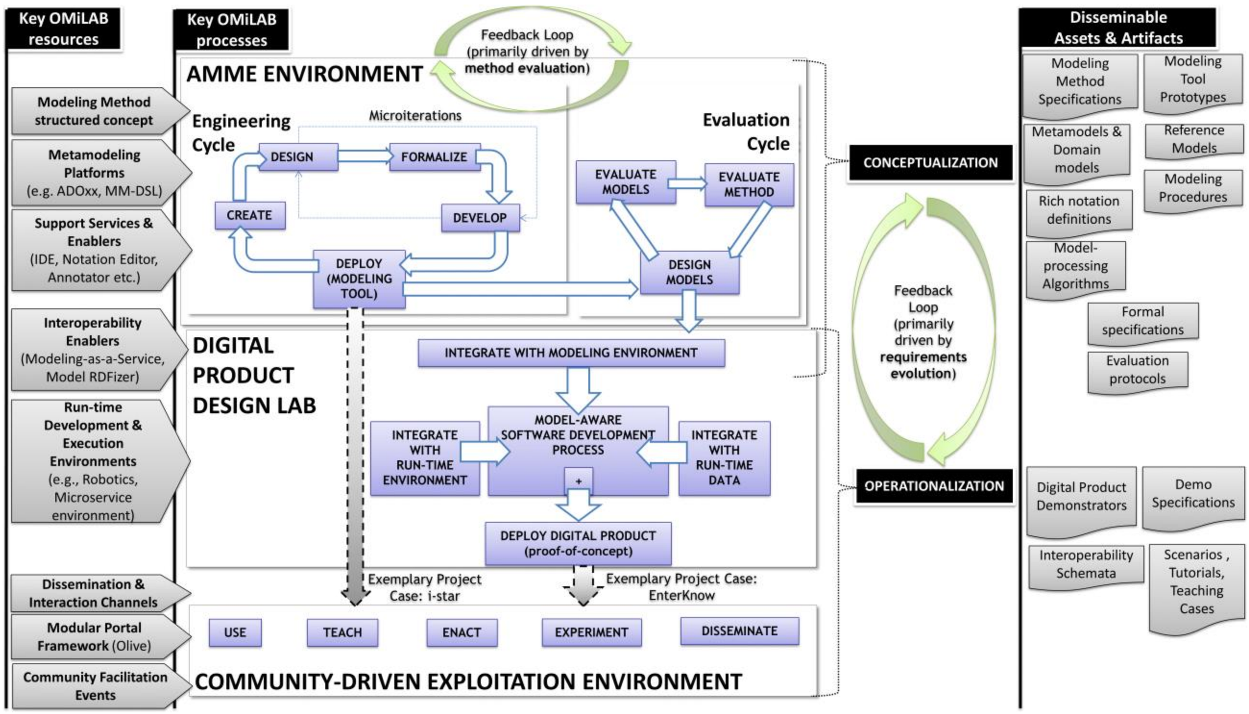

Figure 3. OMiLAB Conceptualization and Operationalization: A Process-centric View

3 Modeling Method Conceptualization in OMiLAB

3.1 The Modeling Method Concept

Conceptual modeling methods help one manage complexity by applying abstraction for a specific purpose. According to Karagiannis and Kühn (2002), a comprehensive modeling method comprises a modeling language, a modeling procedure, and mechanisms and algorithms (see Figure 4). A modeling language forms the core building block of a modeling method, and one can further decompose the language into the syntactic elements (syntax), their graphical representation (notation), and their meaning (semantics). The modeling procedure describes the steps that modelers take and the results they gain from applying a modeling language in order to create valid models. Mechanisms and algorithms define the functionality (e.g., simulation, transformation) that method engineers should implement in a corresponding modeling tool.

One can distinguish domain-specific modeling methods from general-purpose ones based on their pragmatics. The former have the potential to address domain specificity in all their building blocks, while the latter focus on comparability, interoperability, reusability across domains and possibly standardization. As another emergent requirement, modeling methods also require agility in the sense that they can help a method's building blocks (and tool implementation) to evolve and, thus, answer changing modeling needs.

One can further differentiate domain-specific modeling methods from general-purposes ones when considering their application domain. In computer science, modeling methods are designed with the goal of model-driven systems development. Such models often lack proper visualization and focus instead on the capabilities of model transformation and code generation (Moody, 2009). In information science or knowledge management, modelers use modeling methods to develop model-driven systems but also to create abstract representations for “purposes of understanding and communication” (Mylopoulos, 1992, p. 50) (hence, the generalized perspective on model value that we introduce in Section 2).

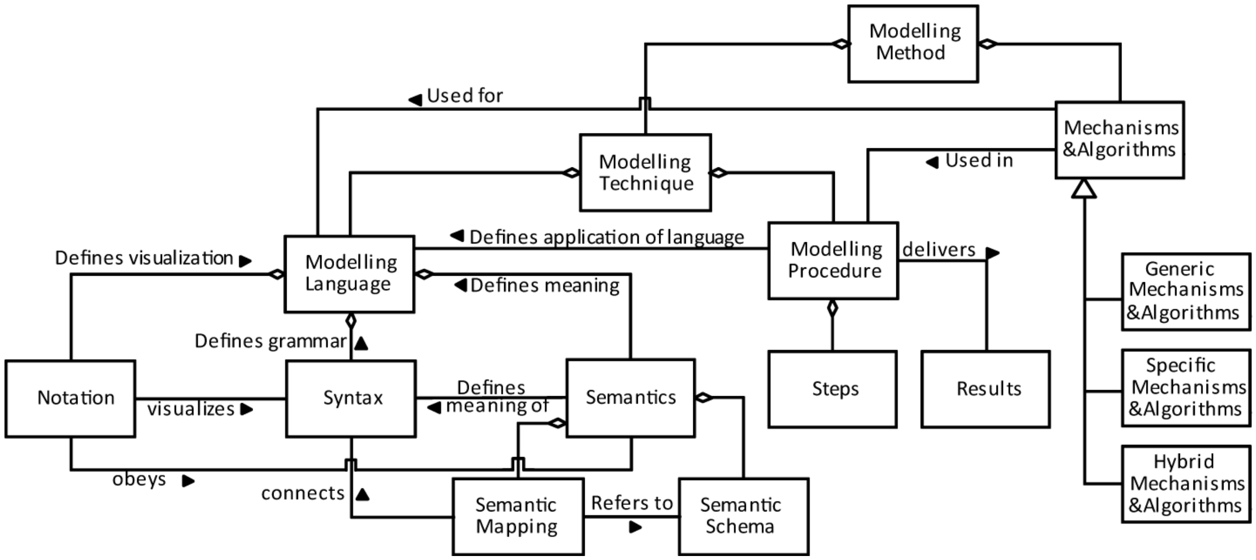

Figure 4. Components of Modeling Methods (Karagiannis & Kühn, 2002)

3.2 The AMME Lifecycle

A modeling method's building block may need to respond to emergent or evolving needs for extension, adaptation, hybridization, or general customization. Rapid prototyping enablers must facilitate iterative and incremental implementation, which moves away from the traditional perception that modeling languages aim for global consensus or standardization. In the OMiLAB's generalized understanding of model value, one may even customize a modeling language for a single enterprise to support some model-aware system or digital product. Therefore, we need to complement the structural view that we provide in Section 3.1 with a dynamic view that considers the specific engineering phases and intermediate artifacts produced during an agile modeling method's lifecycle.

The agile modeling method engineering (AMME) framework (Karagiannis, 2015) prescribes the iterative lifecycle that Figure 3 presents. The AMME framework covers an engineering cycle that starts with knowledge acquisition and requirements analysis (the *create* phase) and ends with the deployment of a usable modeling tool (the *deploy* phase). The intermediate phases include *design* (which results in a specification of the modeling method building blocks), *formalize* (which results in a formalism-oriented specification), and *develop* (which results in an implemented modeling tool). Method engineers can apply microiterations between the *design* and *develop* phases to apply quick changes while skipping the other phases. Specific methodologies, tools, and best practices that the OMiLAB portal hosts supports each phase (see Section 4.3). The ADOxx metamodeling platform (ADOxx.org, 2018; Efendioglu, Woitsch, & Utz, 2016), which provides built-in functionality (e.g., model management, user interaction) that allows method engineers to focus on the modeling method's building blocks without being distracted by generic programming concerns, represents the most prominent resource. This platform significantly improves productivity as developers will spend most of their effort on (see Bork & Sinz, 2010): 1) defining the metamodel by instantiating ADOxx's meta-metamodel concepts, 2) defining visualization and linking concepts across language partitions (model types), and 3) implementing functionality (model transformations, model queries, simulations, etc.).

The iterative and incremental nature of the AMME framework's engineering cycle enables agility because the framework expects modeling requirements to evolve similar to how agile software development principles emerged from a need for responsiveness and manageable granularity in software engineering. A feedback loop occurs between this engineering cycle and an evaluation cycle based on hands-on experience with the deployed modeling prototype as Figure 3 suggests.

3.3 The Digital Product Design Lab

Most projects that the OMiLAB currently hosts deploy a modeling tool and commonly include some model-aware functionality to maximize model value in various decision-support contexts (e.g., simulation, transformation, model queries). However, some projects advance further to the last phase that Figure 2 depicts: engineering model-aware digital products that benefit from resources of the digital product design lab component of the OMiLAB. Typically, these resources are derived by a generalization from past OMiLAB projects such as the microservice architecture for interacting with robots (Walch & Karagiannis, 2017), a plug-in for serialization of models as RDF knowledge graphs (Karagiannis & Buchmann, 2018), consistency management using semantic queries (Karagiannis, Buchmann & Bork, 2016a), or a “model base-as-a-service” architecture that ADOxx provides. The modeling method that one develops though the AMME framework may take its increment requirements (and also modeling concepts) from the elements of a digital product design framework in which the modeling method acts as a “knowledge engine” on the conceptual modeling layer as Figure 5 depicts.

Consequently, two contextual facets determine modeling method conceptualization: 1) the *application domain* (through the “business layer”) where some high-level conceptual framework (e.g., design thinking, Zachman framework) typically provides an initial set of concepts that one refines through the AMME lifecycle and 2) the *IT deployment* (through the “proof-of-concept layer”) whose technology-specific constructs one must map to modeling concepts. These concepts possibly impose constraints on and the need for interoperability between the modeling environment and the digital product’s run-time environment.

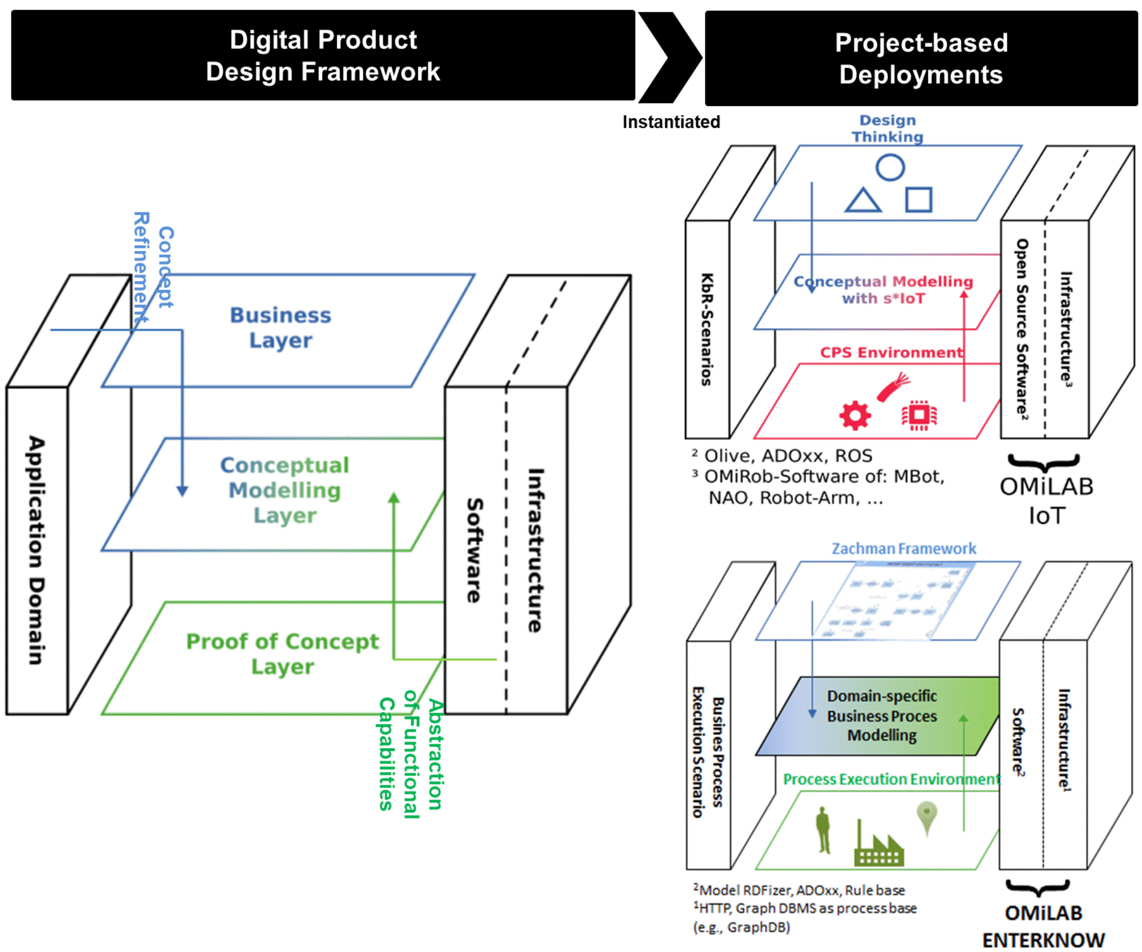

Figure 5. The Digital Product Design Framework

Concepts and properties of the “conceptual modeling layer” may emerge from the “business layer” by refining business concepts (i.e., applying specialization and decomposition), or from the “proof-of-concept layer” by abstracting some functional capabilities of the underlying execution environment. One can find some generalized interoperability mechanisms in past project instances developed in the OMiLAB as Figure 5 suggests (e.g., the plug-in for exporting any kind of model to resource description framework (RDF) knowledge graphs). Figure 5 includes the following project examples:

- A knowledge-based robotics (KbR) project that demonstrates the coordination and operation of robotic arms through diagrammatic models that prescribe behavior, environmental layout, artifacts, and rules (Walch & Karagiannis, 2017, 2019). The deployment employs design thinking as high-level methodology and cyber-physical systems as a low-level execution environment. It also includes an IoT-specific modeling tool that establishes the conceptual bridge across these layers and a microservice-based interoperability framework that enables execution.
- The Enterknow project, which realizes a semantic business process management system that bridges the refined Zachman Framework (Zachman, 1987) concepts from the business layer to machine-readable knowledge graphs that can be executed in a semantics-driven system on the proof-of-concept layer. In Section 6.2, we discuss this project in more detail to better highlight how the OMiLAB conceptualization process extends towards developing model-aware digital product prototypes.

4 The OMiLAB Open Innovation Community

The open source movement inspired and the belief that the conceptual modeling community has not yet harnessed its collaborative innovation potential (which the “reference model” notion (Kock, Strecker, & Frank, 2006) foreshadowed) motivated the OMiLAB concept. Successful communities tend to center around a technological and dissemination platform, which enables the collective and individuals to contribute their ideas. An initial feasibility study (Karagiannis, Grossmann, & Höfferer, 2008) assessed the prevalence of modeling editors and tools and the need for an open innovation community in conceptual modeling.

Founded in 2008, the OMiLAB community has continually evolved. Pioneering projects include the business engineering navigator (Winter, 2010), the semantic object model (Bork & Sinz, 2010; Ferstl, Sinz, & Bork, 2016), and the exemplary business process modeler (Breitling & Hofer, 2012). As at 2019, the OMiLAB community has successfully conceptualized almost 50 different modeling methods that originate from heterogeneous domains, such as multi-perspective enterprise modeling (MEMO) (Bock & Frank, 2016), design thinking (Hawryszkiewycz & Prackwieser, 2016; Bork, Karagiannis, & Hawryszkiewycz, 2017; Miron, Muck, Karagiannis, & Götzinger, 2018), and knowledge management (Cairó & Guardati, 2012; Battistutti & Bork, 2017). Recent conceptualizations target bridging design thinking and conceptual modeling (Miron et al., 2018; Miron, Muck, & Karagiannis, 2019) and industrial business process management (Utz & Falcioni, 2018). Karagiannis et al. (2016c) comprehensively surveys successful conceptualizations, and the website http://austria.omilab.org provides a full project repository.

4.1 Collaboration Possibilities

Besides individuals who casually visit the OMiLAB portal (see Section 4.3) to download a modeling tool for their own needs without any other interaction, the OMiLAB involves structured collaboration that prescribes three commitment levels: member, associated partner, and operator (of an OMiLAB node).

Members (mostly individuals) contribute with their domain knowledge and modeling method to the OMiLAB in order to conceptualize a modeling prototype to help them experiment for a publication or thesis. They must make available an open modeling tool for their method. Research groups can become associated partners (APs) of the OMiLAB, which means they take on additional commitments to contribute to OMiLAB events and to promote OMiLAB resources (e.g., in university courses). Therefore, APs form the core forum that drives OMiLAB's international visibility and publicizes the resources (tools, knowledge assets, methods) that emerge from the community. Table 2 depicts the different adoption levels of OMiLAB resources in a university where a business information systems study program employs OMiLAB resources for various topics across the provided study levels.

Finally, the operator status implies the willingness to establish the physical infrastructure and knowledge that one needs to train others and to host OMiLAB resources. Currently, two physical OMiLABs exist that ensure regional collaboration facilities: one at the University of Vienna in Austria and the other at the University of Chonbuk in Korea. Both labs focus on becoming the primary address for method engineers and tool developers in their respective region. A third OMiLAB in Europe has entered the final preparation phase. Such distributed physical laboratories enable agility in the OMiLAB’s administrative and managerial aspects.

The various freely available modeling tools in Karagiannis et al. (2016c)—the first volume of a joint publication series from the OMiLAB community—document the OMiLAB community's results and the international distribution of its collaborators. Periodic community events are also essential in fostering collaboration and joint publications. The OMiLAB's history[1] covers events with both educational and research purposes in the form of an annual summer school and thematic workshops affiliated with highly visible conferences, respectively.

Table 1. Exemplary Adoption of the OMiLAB in a Partner University

	Topics	**OMiLAB Resources**
Levels >=4: PhD, post doctorate, project-based research	Semantic technology and conceptual modeling (with business administration as application domain)	OMiLAB as hosting and dissemination environment. AMME as design science framework. Metamodeling platforms (e.g., ADOxx) and reusable artifacts. Community-driven domain knowledge and lessons learned.
Level 3: master's—advanced topics	Enterprise modeling, semantic business process management	Free modeling tools for 4EM, ArchiMate, domain-specific tools. Intermediate deliverables of modeling method specifications and modeling tool implementations
Level 2: master's—early topics	Business process management systems, knowledge management systems	Free modeling tools for BPMN, EPC, Petri Net (e.g., Bee-Up)
Level 1: bachelor	Database design, software engineering	Free modeling tools for UML, ER (e.g., Bee-Up)

4.2 OMiLAB in Education

Initially founded through the Erasmus+ Strategic Partnership Project OMI-KA2, the annual Next-generation Enterprise Modeling (NEMO) Summer School[2] , one key initiative in the OMiLAB, enables a comprehensive evaluation environment for modeling methods and educational materials. Since 2014, distinguished professors from different continents with varying domain backgrounds present their view on next-generation enterprise modeling to an international cohort of master's and PhD students. The NEMO program comprises theoretical lectures and practical exercise sessions in which students apply conceptual modeling in emerging domains such as Smart Cities (Bork et al., 2015, 2016). NEMO enables the geographically dispersed community to have an annual meeting, to share experiences, and to discuss future collaboration opportunities. Students gain the unique opportunity to interact with professors and other students from different cultural and educational backgrounds and to follow a highly advanced short-term training program in various aspects of conceptual modeling.

The Bee-Up tool[3], one key educational resource, supports courses on the foundations of modeling with well-established languages such as BPMN, EPC, ER, UML, and Petri Nets. Besides that, the OMiLAB encourages community members to share training material for their domain-specific methods, which allows the community to test and experiment with a broad and diverse audience and, thus, to refine evaluation protocols that provide valuable input for scientific publications.

Finally, the OMiLAB aims to establish valuable training resources for tool developers and digital product engineers in order to facilitate productive prototyping based on scientific principles, such as how to design meaningful and intuitive graphical visualizations (see Moody, 2009; Stark, Braun, & Esswein, 2017; Bork et al., 2018a), how to specify metamodels (Bork, Karagiannis, & Pittl, 2018b), or how to specify mechanisms & algorithms that process the model semantics (Buchmann & Karagiannis, 2016; Buchmann & Karagiannis, 2017).

4.3 The OMiLAB Portal

The OMiLAB Portal constitutes the central space for OMiLAB dissemination and resource hosting. One can openly access it at www.omilab.org to present, for example, modeling projects, events, services, and tools. Based on Olive[4], an internally developed content management system deployed in a flexible

[1] See http://www.omilab.org/psm/content/ep/globalnetworkservice?view=tilesevents
[2] See http://nemo.omilab.org
[3] See http://austria.omilab.org/psm/content/bee-up/info
[4] See https://www.adoxx.org/live/olive

microservice architecture, each OMiLAB project can disseminate information about the domain requirements, use cases, tutorials, training materials, a wiki, and download pages for its prototypes (e.g., modeling tools) in a structured way.

The portal also hosts a variety of tools and services that complement the metamodeling platform—typically outcomes of past projects that have been refined to be reusable beyond their original context:

- Users can evaluate the integrity of OMiLAB models with the Semcheck project (Jeusfeld, 2016) using the ConceptBase system (Jarke, Gallersdörfer, Jeusfeld, Staudt, & Eherer, 1995).
- The GraphRep Generator helps users to design the notation for a modeling language's elements in an intuitive way by providing a what-you-see-is-what-you-get drawing editor. Users can draw elements and drag and drop existing composed elements into a drawing area. When they finish the design, they can automatically generate ADOxx-specific code from the drawings.
- Users can use MM-DSL (Visic et al., 2015) to specify a modeling language's constituents via a declarative language and then compile the source code to platforms such as ADOxx in order to generate a modeling tool.
- An OMiLAB IDE supports the modeling tool-development phase in the AMME framework with code completion, debugging, and features already known from IDEs such as Eclipse.
- Any modeling tool that users develop on ADOxx can export its diagrams to RDF graphs with the help of an external plug-in that Karagiannis and Buchmann (2016) introduced to support 1) interoperability with model-aware digital products and artifacts and 2) reasoning and semantic queries across multiple related models or model-data mash-ups.

5 Selected Application Cases

Typical OMiLAB project goals range from 1) producing strictly a modeling tool for specific requirements or scientific experimentation to 2) implementing a digital product that employs such a tool as a knowledge-representation instrument. In this section, we illustrate the two categories with two exemplary selected projects: one tailored for requirements engineering (iStar; see Section 5.1) and one showcasing model-aware semantic information systems (EnterKnow; see Section 5.2). We refer back to Figure 3 that indicates which path through the overall conceptualization process each project takes.

In discussing each example, we highlight the project-specific requirements and their conceptualization as modeling methods in the OMiLAB digital ecosystem. We introduce each case through the method-specific requirements and means to address them.

5.1 The iStar Case

The iStar language initially focused on modeling strategic relationships for process re-engineering by focusing on intentional, social, and strategic dimensions (Yu, 1995). In recent years, researchers have proposed several ways of extending and adapting iStar, iStar model-analysis techniques, and iStar modeling tools (see Horkoff & Yu, 2016; Li, Grubb, & Horkoff, 2016). As iStar evolves, agility at modeling method level becomes a necessity, which makes it an ideal case for the AMME framework. In this section, we present the current conceptualization of iStar in the OMiLAB[5] (Franch, López, Cares, & Colomer, 2016). The iStar requirements cover not only notational aspects but also processing functionality in the modeling environment, such as:

1) Boundary highlighting: the notation needs to intuitively encode the relationships between intentional elements and intentional actors.
2) Semi-automated dependency definition: the tool should intuitively support dependency creation.
3) Goal satisfaction evaluation: the goal satisfaction shall be computable.
4) Intentional actor relationships: modelers should have the possibility to overview the relationships of one or more intentional actors.

[5] Available online at http://austria.omilab.org/psm/content/istar/info

5) Interpretability: strategic dependency model and strategic rational model should be interpretable.

For the first requirement, intentional actors may visualize a boundary to express that they explicitly desire all intentional elements in this boundary. If modelers place no actor in proximity or in a boundary, the tool shows a hint to help them create semantically valid iStar models. In order to support how modelers define dependencies (second requirement), the iStar tool provides a drop-down dialog whenever the modeler has created a new dependency relation. The modeler may use this drop-down to select the correct type of dependency. The tool then automatically places a corresponding element in the dependency relation. Having defined the satisfaction of all goal elements in an iStar model, the modeler may execute an algorithm that traverses the goal hierarchy and computes an overall goal satisfaction value (third requirement). An algorithm that runs with one or more models and generates a matrix for all intentional actors (depender, dependum) with their relationship status (committed, open, critical) realizes the fourth requirement. For the fifth requirement, the tool uses the *view* mode of ADOxx, which enables the modeler to efficiently switch between the strategic dependency view and strategic rational model view (Schwab, Karagiannis, & Bergmayr, 2010).

5.2 The EnterKnow Case

The EnterKnow project[6] has developed a business process management system (BPMS) whose front-end behavior a back-end hybrid knowledge base determines at run time. The knowledge base semantically integrates several information sources: model contents, run-time data, and open geographical data that one can access through various services (e.g., Google Maps[7]). The system integrates these sources through linked-data techniques (Heath & Bizer, 2011) in a common graph database, which results in hybrid knowledge graphs that reasoning mechanisms (OWL axioms and GeoSPARQL-based rules[8]) extend further.

A domain-specific business process modeling language tailored to incorporate concepts from the who, how, and where facets of the Zachman framework (Zachman, 1987) forms the core of this complex artifact. The system exposes model contents that capture knowledge across these dimensions to the end user's run-time software through the interoperability plug-in that exports model contents in the RDF format for linking and reasoning (e.g., OWL axioms, GeoSPARQL). The software engineering method that this project adopts (see right side of Figure 6)—which Buchmann, Cinpoeru, Harkai, and Karagiannis (2018) introduced and analyzed—instantiates the OMiLAB's model-aware software-development process (see Figure 3) and uses RDF as an interoperability format and GraphDB's REST API[9] as an interoperability channel.

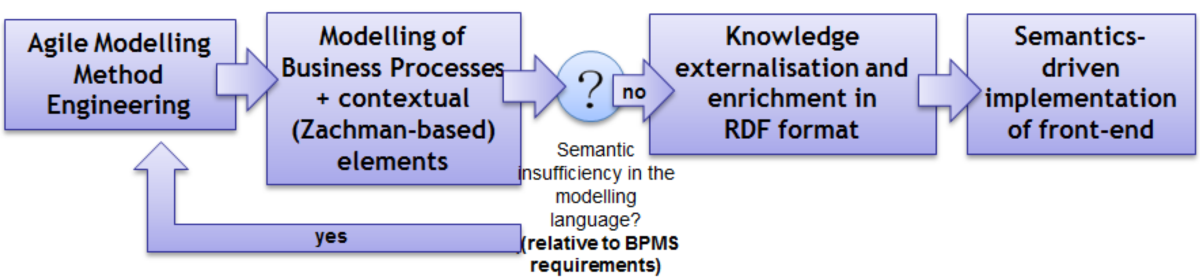

Figure 6. The EnterKnow Model-aware Software Engineering Process

We can summarize the following specific requirements for EnterKnow:

1) To extend a business process modeling language with several perspectives that the Zachman framework recommends (more precisely, their organizational context ("who") and geographical locations ("where")).
2) To use openly available geographical data (and potentially any other linked open data that one can connect to model elements in a meaningful way).

[6] See http://austria.omilab.org/psm/content/enterknow/info?view=home
[7] See https://cloud.google.com/maps-platform/
[8] See http://www.opengeospatial.org/standards/geosparql
[9] See http://graphdb.ontotext.com

3) To minimize the inherent conceptual redundancy between the data model that a BPMS's database employs at run-time and the metamodel that governs the modeling tool at design-time.
4) To enable one to integrate the model contents and run-time data in a way that is amenable to reasoning across the integrated model and data knowledge sources.

The EnterKnow prototype satisfies the third requirement by assimilating, in the modeling language agilely tailored for the first requirement, parts of the data model that a BPMS employs at run-time. By doing so, the project partially fills the semantic gap that inherently exists between traditional model-driven software and the standard modeling languages that provide an invariant semantic space. Thus, the novel approach empowers modelers with the ability to apply diagrammatic configurations for software artifacts' run-time behavior. The project reconsiders the traditional round-trip engineering cycles through the lens of semantic technology and machine-readable knowledge flows.

The project satisfies the second and fourth requirements with the help of linked data-based interoperability mechanisms that Karagiannis and Buchmann (2018) detail:

- An ADOxx RDFizer plug-in and an RDF vocabulary for exporting any diagrammatic model. Karagiannis and Buchmann (2016) discuss transformation patterns to support generalization and potential adaptation for other metamodeling platforms. The RDFizer becomes an adapter for semantic lifting, which contributes to a growing list of such adapters that researchers have developed for legacy information sources (e.g., Langegger & Wöß, 2009). As such, the OMiLAB offers the RDFizer as a reusable resource. Researchers have also reported earlier domain-specific applications for tracing mobile application requirements and for enriching queries in the Internet of Things (see Buchmann & Karagiannis, 2016; Buchmann & Karagiannis, 2017; Buchmann, 2016).
- Mechanisms for establishing semantic links between diagrammatic elements and external resources of various kinds: terms from existing vocabularies or ontologies, data entities from external linked open data repositories, the possibility to attach (to any diagrammatic element) arbitrary RDF statements that the modeling language (i.e., the metamodel) does not restrict.

These mechanisms enrich the AMME framework with additional steps to achieve a full-fledged software development method (see Buchmann et al., 2018) or, if approached from a knowledge management perspective, a knowledge conversion cycle that supports the development of knowledge-driven digital products (Karagiannis, Buchmann, & Walch, 2017).

6 Related Approaches

Method engineering has a long tradition in information science, computer science, and systems development (see Brinkkemper, 1996; Nuseibeh, Finkelstein, & Kramer, 1996; Ralyté, Deneckère, & Rolland, 2003; Rolland, 2009). In recent years, researchers have also viewed it as an important area of IS research (Recker, 2015) that focuses on developing domain-specific conceptual modeling methods (Frank, 2013; Frank et al., 2014). Due to its ambiguous usage (Henderson-Sellers & Ralyté, 2010), one needs to tailor method engineering needs to the notion of conceptual modeling and model value that the OMiLAB employs (see Figures 2 and 3).

In contrast to the more general term “method engineering” in systems engineering (where users establish situation-specific methods by combining existing method chunks), in conceptual modeling, researchers understand method engineering as the process of creating a new modeling method to create conceptual models that abstractly describe some real-world phenomenon to foster understanding and support communication (which we extend to “interoperability”) as Mylopoulos (1992) has proposed. The conceptualization process that the OMiLAB digital ecosystem enables (see Figure 3 at the top) allows one to create modeling methods in such a way: the ecosystem's building blocks focus on designing new conceptual modeling methods that align to domain- or case-specific requirements. Moreover, the OMiLAB assists one in operationalizing outcomes with help from specialized kinds of resources and services that its digital ecosystem makes available (see Figure 3 at the bottom).

Besides the primary scope of the OMiLAB as we describe above, approaches that focus on establishing open communities in method engineering and conceptual modeling fields relate to the OMiLAB. For example, Koch et al. (2006) built a community for producing and sharing reference models for software development. Other researchers have established several other community-oriented initiatives in recent years as well. Table 3 brief describes the most relevant ones and contrasts them to the OMiLAB.

Table 3. Overview of Related Approaches on Method Engineering and Community Building in IS

Approach	Description
DSM forum	"The DSM forum exists to spread the knowledge and know-how of domain-specific modeling. It is an independent body made up of the leading DSM tool and solution providers, along with expert DSM users"[10]. The DSM forum homepage lists commercial and open DSM tools, DSM publications, and DSM case studies. The forum does not restrict users to a certain application domain such as conceptual modeling or model-driven engineering. Moreover, it does not restrict users to a certain modeling tool or development platform. The DSM forum focuses on providing content that specifically benefits researchers and practitioners. Consequently, it does not provide any learning material or training. Moreover, the DSM forum does not feature an active community: it merely serves as a repository for relevant information and technologies useful for conceptualizing modeling methods.
Free Models Initiative	A kick-off workshop organized at the Modellierung conference established the Free Models Initiative in 2014. The initiative has a vision to provide free model corpora for researchers in the software engineering, model-driven development, and business process management domains. The corpora shall enable scientific benchmarks, evaluations, and research model-based approaches. The initiative does not provide modeling tools, nor does it address educational aspects. The initiative homepage lists existing model corpora. This initial activity should lead to a living community. Despite the interesting and relevant goal, we found that the initiative already lacked activity at the time we wrote this paper. One can assert that this initiative focuses on providing open models for an open community. However, in contrast to the OMiLAB, this initiative does not focus on defining new modeling methods or on realizing and providing modeling tools. Lastly, the initiative focuses on providing model corpora for research purposes.
GenMyModel	In January, 2013, the GenMyModel initiative started to establish a worldwide community of software engineers (Dirix, Muller, & Aranega, 2013). The platform provides basic functionality to collaboratively create models relevant for software engineering. The platform currently supports the following modeling languages: ArchiMate, BPMN, DMN; UML, RDS, and Flowchart. GenMyModel's strength lies in its low entry barrier for modelers by its browser-based online modeling and collaboration environment. The platform comes with a free sign-up package (limited model size, limited collaborators, etc.), but, for reasonable usage, users require a monetary subscription. Users can share and discuss the created models in the community. In contrast to the OMiLAB, this approach fosters on the modeling part of de facto industry standards built on an economic business model. Di Rocco, Di Rocco, Iovino, and Pierantonio (2015) and Popoola, Carver, and Gray (2017) overview similar modeling-as-a-service approaches for software engineering.
WeST	The Institute for Web Science and Technologies was founded in 2009. It focuses on establishing a multi-perspective research framework on topics related to the digital transformation. Thereby, it focuses on Web services and Web technologies such as semantic Web, data mining, interactive Web, and software and services in the Web. Additional working groups focus on digital markets, e-government, and digital entrepreneurship. It covers an impressive spectrum of topics. Besides, WeST also incorporates a huge number of international industry and research partners. Research mostly occurs in research laboratories in Europe with facilities also in North and South America and Asia. WeST acquires a lot of funding budget for research projects, mostly in the German-speaking and European area. WeST also focuses on education. Since 2016, it has organized an annual WSTNet Web Science Summer School. Moreover, the University of Koblenz-Landau offers a Master's Pin Web Science in cooperation with the WeST. WeST has a strong focus on applied research that provides a benefit for the industry. It does not employ or share a set of technologies. Users require expert knowledge in Web technologies and semantic technologies to contribute to the community. It is not clear how interested researchers can collaborate.

To the best of our knowledge, there is no initiative comparable to the OMiLAB that has an identical scope, shares a similar vision of openness (Bork & Miron, 2017), and provides a rich set of enablers. OMiLAB focuses on a co-creation strategy and its required digital ecosystem for conceptualizing modeling methods. All the above initiatives have their strengths in isolated aspects but do not provide a holistic view

[10] See http://www.dsmforum.org/

about conceptualizing and operationalizing conceptual modeling methods. We see this research also as a means to start collaborating with those initiatives in order to bundle their strengths and to bridge the information and computer science modeling communities.

7 Evaluation

The OMiLAB comprises a multitude of tools, services, best practices, and a repository of reusable tools and knowledge assets that pertain to the conceptualizing and operationalizing modeling methods. The case studies we present in Sections 5.1 and 5.2 show the OMiLAB's feasibility for conceptualizing modeling methods. In this section, we report two additional evaluations. First, we provide quantitative measures for 1) the OMiLAB's practical impact by referring to OMiLAB modeling tool downloads and 2) its educational impact by referring to the Next-generation Enterprise Modeling Summer School. Second, we use lessons learned from a failed attempt to establish an open ecosystem (von Briel & Recker, 2017) to evaluate the extent to which the OMiLAB is prepared for such lessons.

7.1 Quantitative Evaluation

7.1.1 Practical Impact

Using the OMiLAB digital ecosystem, research groups around the world have realized almost 50 modeling methods (Karagiannis et al., 2016c). The OMiLAB portal hosts all methods and the corresponding modeling tools (see Section 4.3). The methods are tailored for a variety of (both educational and research-oriented) project-based use cases and domain-specific requirements

We evaluate the OMiLAB's practical impact by looking at the download numbers for OMiLAB modeling tools. In July, 2016, the OMiLAB deployed its portal, which has operated since. As part of this portal, a file manager microservice handles all file downloads, which includes modeling tool downloads. In this 29-month period, the portal recorded 8,277 downloads (see Figure 7). Consequently, every month, users downloaded approximately 285 OMiLAB tools and presumably used them in education or research. The peaks in March indicate that educators likely used the tools in university courses (most European universities start their summer semester in March/April). The peak in July, 2018, likely indicates that NEMO students downloaded the tools that corresponded to the lectures. As at November, 2018, 18 modeling tools had seen more than 100 downloads each, and 31 tools had seen more than 30 downloads each.

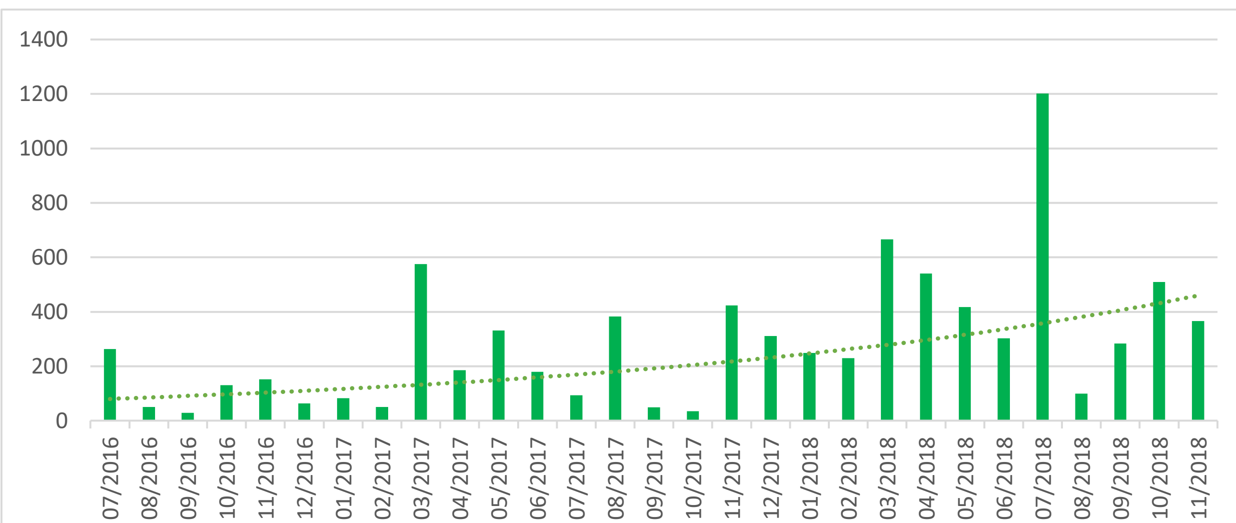

Figure 7. OMiLAB Tool Downloads from July, 2016, until November, 2018

Note that these numbers establish the lower bound of the actual download and usage numbers. Community members can obviously also upload their tools and provide them to their students via local learning management systems such as Moodle or via private webpages. Thus, the actual practical impact is likely even higher. The OMiLAB encourages collective intelligence for the common good (see Schuler, De Cindio, & De Liddo, 2015). As the realized tools gain maturity and the methods publicity, the OMiLAB network establishes a knowledge base for modeling methods, open tools, and freely available teaching materials.

7.1.2 Educational Impact

As for the OMiLAB's educational impact, OMiLAB resources cover multiple abstraction layers and application domains for teaching conceptual modeling as a self-contained field and in a consistent manner. By subordinating modeling practices to other fields, one tends to fragment and reduce conceptual models' general value to isolated (albeit popular) use cases. The OMiLAB stimulates a top-down holistic view that one may further specialize across study programs for their relevant application domains. As a quantifiable measure for the OMiLAB's educational impact, we refer to the NEMO Summer School series. In its first five years, NEMO featured 197 theoretical lectures and numerous practical exercises. In total, the program has educated 284 students in manifold aspects of conceptual modeling (see Table 4).

Table 4. Students and Lecturers of the NEMO Summer School Series

		2014	2015	2016	2017	2018
Students	Students	50	55	67	57	55
	Institutions	29	26	43	39	38
	Countries	15	16	28	25	25
Lecturers	Lecturers	33	38	43	45	38
	Institutions	26	32	35	33	31
	Countries	15	20	19	18	19

A detailed qualitative survey among 75 students in NEMO 2016 (see Bork et al., 2016) and 2018 shows that students appreciated teaching conceptual modeling in interesting domains such as smart cities, which forms the base for one core exercise in recent NEMO editions and also part of the OMiLAB training material on metamodeling. Students appreciate using open tools in the practical sessions as a means to gain hands-on modeling experience. Further, the survey results show that 72 of the 75 students mostly agreed or fully agreed that they would recommend participating in future NEMO editions to their colleagues or other students.

7.2 Literature-based Lessons Learned

We relate to von Briel and Recker (2017) who identify pitfalls and fallacies and synthesize a set of lessons learned for online open innovation communities. In this section, we ground those lessons learned in the case of the OMiLAB.

7.2.1 Lesson 1: Acknowledge the Industry Context

Acknowledging the industry context currently represents only a marginal concern since the OMiLAB targets primarily academic researchers and lecturers who embrace the philosophy of open use, open access, and open source and who want to spread their ideas and increase the visibility of their projects. Industry stakeholders typically become indirectly involved in projects as project partners; however, projects and their contributors (who each decide what to contribute according to their project-level agreements) retain the intellectual property for OMiLAB contributions. In NEMO 2018, we actually welcomed participants from industry for the first time.

A non-profit organization manages the OMiLAB and fosters it as a public good to facilitate the state-of-the-art enrichment and to push a teaching agenda for conceptual modeling as a mature field that produces specific kinds of value. As industry stakeholders may show more active involvement, the organization could plausibly raise requirements to partition the community in different sections. The in-house content management system underlying the OMiLAB portal can handle modularity and extensibility for such an evolution.

7.2.2 Lesson 2: Consider the Legal and Regulatory Environment

The OMiLAB aims to grow as an international network of OMiLAB installations (e.g., OMiLAB Korea) that provide geographical coverage and adaptation to regional regulatory environments. Currently, the OMiLAB stores its data in Europe and has begun to refine policies to comply with GDPR regulations. The

OMiLAB non-profit organization manages the infrastructure and administrative aspects that pertain to OMiLAB events.

7.2.3 Lesson 3: Establish Support Processes Early

The OMiLAB develops support on a continuous basis in the form of services that facilitate various conceptualization steps (e.g., design of graphics, model annotation, plug-ins, IDEs), and tutorials and demos of varying specificity to train users on each conceptualization step in the AMME framework. The OMiLAB portal adopts a microservice content management system architecture that allows users to quickly set up new project spaces with adequate granularity and extensible sections. Dedicated personnel maintain this architecture to ensure the OMiLAB responds to community members' needs.

Also, the AMME framework and the modeling method seed concept adopt a modular way to avoid heavyweight and rigid implementation processes, which allows contributors to become involved in the steps where they hold most interest (e.g., picking an open method specification and deploying it, replacing the notation of a modeling language, integrating Web services and third-party tools).

7.2.4 Lesson 4: Prepare to Shift the Organizational Mindset and Lesson 5: Get Ready to Adapt Your Current Innovation Approach.

We cannot yet assess how this digital ecosystem has evolved since it has not existed for long enough, but it should do so in line with how conceptual modeling evolves as a research paradigm with milestones such as Sabegh, Lukyanenko, Recker, Samuel, and Castellanos' (2017b) study. Such studies gradually define requirements that will motivate the OMiLAB's future strategies.

7.2.5 Lesson 6: Know Your Contributors

OMiLAB stakeholders meet frequently either in dedicated OMiLAB events (the NEMO Summer School series, OMiLAB-focused workshops attached to various conferences) or in community-specific scientific conferences (typically those that touch on conceptual modeling topics: ER, PoEM, CAISE, MODELS, etc.). The OMiLAB also hosts project-based research visits on a regular basis at its physical space at the University of Vienna, which various contributors facilitate. Each project that the OMiLAB portal has hosted has included experience exchanges between the OMilAB and the contributors in order to refine the scope and requirements for the OMiLAB.

8 Conclusion

The Open Models Laboratory (see http://omilab.org/) is a digital ecosystem for the conceptualizing and operationalizing modeling methods. It combines open source and open community principles to foster cross-domain practice-oriented information science and design science research. One can find a repository of all realized modeling tools online[11]. The methods and tools that the OMiLAB provides enable research, application, and education in conceptual modeling. Moreover, the OMiLAB allows researchers to deploy their method artifacts in order to publish their proofs-of-concepts and have others evaluate them.

In this paper, we introduce the OMiLAB's fundamental pillars: 1) a granularly defined modeling method concept whose building blocks one can customize for the domain of choice, 2) the agile modeling method engineering framework that helps one quickly prototype modeling tools, 3) a model-aware digital product design lab, and 4) dissemination channels for reaching a global community. Based on two application cases, we demonstrate the OMiLAB's viability. Moreover, we comprehensively quantitatively evaluate the OMiLAB to highlight its practical and the educational impact. The OMiLAB acts as a successful best practice in how to establish an online innovation community. Note that the conceptual contributions we present in this paper (i.e., the OMiLAB digital ecosystem with how it conceptualizes (Figure 3 top) and operationalizes (Figure 5) modeling methods) already have physical installments (as Figure 8 shows).

[11] See http://austria.omilab.org/psm/exploreprojects?param=explore

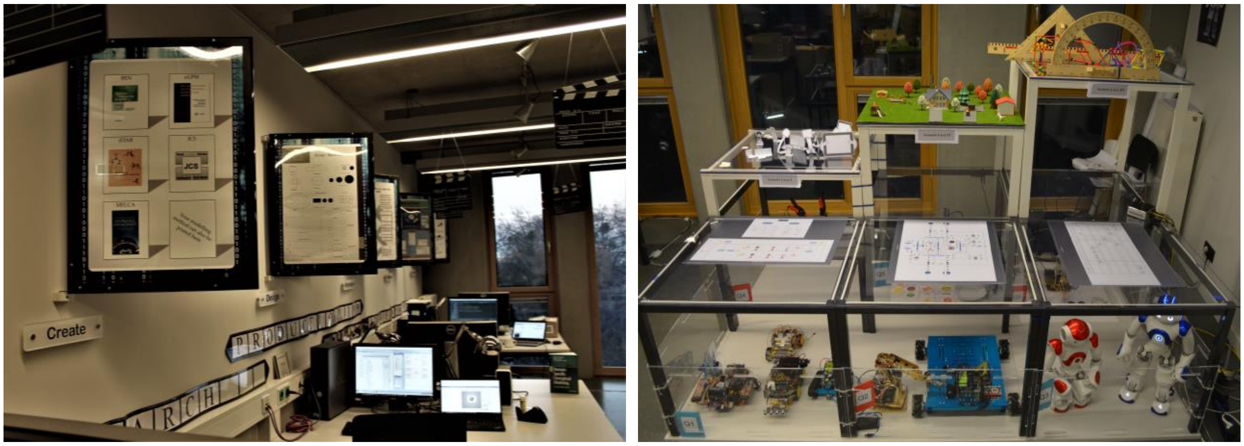

Figure 8. Physical Installments of the Conceptualization (Left) and the Digital Product Space (Right)

Generally, we can differentiate three OMiLAB usage scenarios:

- One can find a modeling method in OMiLAB that suits all requirements. In this case, one can simply **download and use** the tool.
- An existing method in OMiLAB addresses some of the requirements. In this case, one can **extend** the method to fulfill all requirements (see Sabegh & Recker, 2017a).
- No method fits to the requirements. In this case, the OMiLAB can help one **conceptualize** a novel modeling method that specifically addresses the requirements.

By motivating more researchers across the world to contribute their platforms, methods, and tools, we can diversify the ecosystem's technological landscape. Currently, OMiLAB community members have little interest in migrating method specifications across different metamodeling platforms, and no one has documented the challenges derived from such efforts. However, a recent research project has analyzed ADOxx-based metamodels in comparison with Ecore-based ones (Bork, 2018) to support technological openness and migration.

Referring back to our research question, we show the results of our practical work in supporting modeling method engineers. Based on the almost 50 realized OMiLAB projects, we believe that the digital ecosystem provides the necessary tools and services to foster modeling method engineering. However, we need to empirically investigate whether we can further improve the procedure applied and the tools/services. We plan to involve the OMiLAB community in this evaluation process more directly.

Future work will concentrate on further extending the OMiLAB by: 1) involving new community members with their domains (e.g., robotics, industry 4.0), 2) attracting developers of further tools and services supporting modeling method conceptualization in order to improve project success (see Jetu & Riedl, 2012), 3) inaugurating further OMiLAB nodes worldwide to enhance the global network while enabling local bonding, and 4) advocating for the relevance of conceptual modeling in the IS curriculum by establishing NEMO as a key educational forum that propagates ideas and contents towards national-level study programs.

Acknowledgements

The work of Robert Buchmann was supported by the Romanian National Research Authority through UEFISCDI under grant agreement PN-III-P2-2.1-PED-2016-1140.

References

Becker, J., vom Brocke, J. Heddier, M., & Seidel, S. (2015). In search of information systems (grand) challenges. *Business & Information Systems Engineering*, *57*(6), 377-390.

Benlian, A., Hilkert, D., & Hess, T. (2015). How open is this platform? The meaning and measurement of platform openness from the complementors perspective. *Journal of Information Technology*, *30*(3), 209-228.

ADOxx.org. (2018). *ADOxx official page*. Retrieved from http://adoxx.org

Battistutti, O. C., & Bork, D. (2017). Tacit to explicit knowledge conversion. *Cognitive Processing*, *18*(4), 461-477.

Bock, A., & Frank, U. (2016). *Multi-perspective enterprise modeling—conceptual foundation and implementation with ADOxx*. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-specific conceptual modeling* (pp. 241-267). Berlin: Springer.

Bork, D., & Sinz, E. J. (2010). Design of a SOM business process modelling tool based on the ADOxx meta-modelling platform. In J. de Lara & D. Varro (Eds.), *Proceedings 4th international Workshop on Graph-Based Tools (GraBaTs)* (pp. 89-101).

Bork, D., Fill, H. G., Karagiannis, D., Miron, E. T., Tantouris, N., & Walch, M. (2015). Conceptual modelling for smart cities: A teaching case. *Interaction Design & Architectures, 27*, 10-28.

Bork, D., Buchmann, R., Hawryszkiewycz, I., Karagiannis, D., Tantouris, N., & Walch, M. (2016). Using conceptual modeling to support innovation challenges in smart cities. In *Proceedings of the 18th International Conference on High-Performance Computing and Communications, 14th International Conference on Smart City, and 2nd International Conference on Data Science and Systems* (pp. 1317-1324).

Bork, D., Karagiannis, D., & Hawryszkiewycz, I. (2017). Supporting customized design thinking using a metamodel-based approach. In *Proceedings of the 28th Australasian Conference on Information Systems.*

Bork, D., & Miron, E. T. (2017). OMiLAB—an open innovation community for modeling method engineering. In *Proceedings of the 8th International Conference of Management and Industrial Engineering* (pp. 64-77).

Bork, D. (2018). Metamodel-based analysis of domain-specific conceptual modeling methods. In *Proceedings of the 11th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modelling* (pp. 172-187)

Bork, D., Karagiannis, D., & Pittl, B. (2018a). Systematic analysis and evaluation of visual conceptual modeling language notations. In *Proceedings of the 12th International Conference on Research Challenges in Information Science*.

Bork, D., Karagiannis, D., & Pittl, B. (2018b). How are metamodels specified in practice? Empirical insights and recommendations. In *Proceedings of the 24th Americas Conference on Information Systems.*

Breitling, H., & Hofer, S. (2012). Beispielhaft gut modelliert: Exemplarische geschäftsprozess-modellierung in der praxis. *OBJEKTspektrum*, *6*, 8-13.

Brenner, W., Karagiannis, D., Kolbe, L., Krüger, J., Leifer, L., Lamberti, H-J., Leimeister, J. M., Österle, H., Petrie, C., & Plattner, H. (2014). User, use & utility research. *Business & Information Systems Engineering*, *6*(1), 55-61.

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, *38*(4), 275-280.

Bucher, T., Klesse, M., Kurpjuweit, S., & Winter, R. (2007). *Situational method engineering*. In J. Ralyte, S. Brinkkemper, & B. Henderson-Sellers (Eds.), *Situational method engineering: Fundamentals and experiences* (pp. 33-48). Berlin: Springer.

Buchmann, R. A. (2016). Modeling product-service systems for the Internet of things: The ComVantage method. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-specific conceptual modeling* (pp. 417-437). Berlin: Springer.

Buchmann, R. A., & Karagiannis, D. (2015). Agile modelling method engineering: Lessons learned in the ComVantage research project. In J. Ralyte, S. Espana, & O. Pastor (Eds.), *Proceedings of the 8th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling* (pp. 356-373). Berlin: Springer.

Buchmann, R. A., & Karagiannis, D. (2016). Enriching linked data with semantics from domain-specific diagrammatic models. *Business & Information Systems Engineering*, *58*(5), 341-353.

Buchmann, R. A., & Karagiannis, D. (2017). Modelling mobile app requirements for semantic traceability. *Requirements Engineering*, 22(1), 41-75.

Buchmann, R. A., & Ghiran, A. M. (2017). Engineering the cooking recipe modelling method: A teaching experience report. In *Proceedings of the 1st International Workshop on Practicing Open Enterprise Modeling within OMiLAB.*

Buchmann, R. A., Cinpoeru, M., Harkai, A., & Karagiannis, D. (2018). Model-aware software engineering: A knowledge-based approach to model-driven software engineering. In E. Damiani, G. Spanoudakis, & L. Maciaszek (Eds.), *Proceedings of the 13th Int. Conference on Evaluation of Novel Approaches to Software Engineering* (233-240). Setúbal, Portugal: SCITE Press.

Cairó, O., & Guardati, S. (2012). The KAMET II methodology: Knowledge acquisition, knowledge modeling and knowledge generation. *Expert Systems with Applications*, *39*(9), 8108-8114.

Dirix, M., Muller, A., & Aranega, V. (2013). Genmymodel: An online UML case tool. In *Proceedings of the European Conference on Object-Oriented Programming.*

Di Rocco, J., Di Ruscio, D., Iovino, L., & Pierantonio, A. (2015). *Collaborative repositories in model-driven engineering [software technology]*. *IEEE Software*, *32*(3), 28-34.

Efendioglu, N., Woitsch, R., & Utz, W. (2016). A toolbox supporting agile modelling method engineering: ADOxx.org modelling method conceptualization environment. In *Proceedings of the IFIP Working Conference on The Practice of Enterprise Modeling* (pp. 317-325).

Ferstl, O. K., Sinz, E. J., & Bork, D. (2016). Tool support for the semantic object model. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-specific conceptual modeling* (pp. 291-310). Berlin: Springer.

Fill, H.-G., Redmond, T., & Karagiannis, D. (2012). FDMM: A formalism for describing ADOxx meta models and models. In L. Maciaszek, A. Cuzzocrea, & J. Cordeiro (Eds.), *Proceedings of the 14th International Conference on Enterprise Information Systems* (vol. 3, pp. 133-144).

Franch, X., López, L., Cares, C., & Colomer, D. (2016). The i* framework for goal-oriented modeling. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-specific conceptual modeling* (pp. 485-506). Berlin: Springer.

Frank, U. (2013). Domain-specific modeling languages: Requirements analysis and design guidelines. In I. Reinhartz-berger, A. Sturm, T. Clark, S. Cohen, & J. Bettin (Eds.), *Domain engineering: Product lines, languages, and conceptual models* (pp. 133-157). Berlin: Springer.

Frank, U., Strecker, S., Fettke, P., vom Brocke, J., Becker, J., & Sinz, E. J. (2014). The research field modeling business information systems. *Business & Information Systems Engineering*, *6*(1), 39-43.

Galvagno, M., & Dalli, D. (2014). Theory of value co-creation: A systematic literature review. *Managing Service Quality*, *24*(6), 643-683

Hawryszkiewycz, I. T., & Prackwieser, C. (2016). MELCA—customizing visualizations for design thinking. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-specific conceptual modeling* (pp. 383-396). Springer.

Heath, T., & Bizer, C. (2011) *Linked data: Evolving the Web into a global data space* (1st ed.). San Rafael, CA: Morgan & Claypool.

Henderson-Sellers, B., & Ralyté, J. (2010). Situational method engineering: State-of-the-art review. *Journal of Universal Computer Science*, *16*(3), 424-478.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75-105.

Horkoff, J., & Yu, E. (2016). Interactive goal model analysis for early requirements engineering. *Requirements Engineering*, *21*(1), 29-61.

Jarke, M., Gallersdörfer, R., Jeusfeld, M. A., Staudt, M., & Eherer, S. (1995). ConceptBase—a deductive object base for meta data management. *Journal of Intelligent Information Systems*, *4*(2), 167-192.

Jetu, F. T., & Riedl, R. (2012). Determinants of information systems and information technology project team success: A literature review and a conceptual model. *Communications of the Association for Information Systems*, *30*, 455-482.

Jeusfeld, M. A. (2016). SemCheck: Checking constraints for multiperspective modeling languages. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-specific conceptual modeling* (pp. 31-53). Berlin: Springer.

Karagiannis, D. (2015). Agile modeling method engineering. In *Proceedings of the 19th Panhellenic Conference on Informatics* (pp. 5-10).

Karagiannis, D., & Buchmann, R. A. (2016). Linked open models: Extending linked open data with conceptual model information. *Information Systems*, *56*, 174-197.

Karagiannis, D., Buchmann, R. A., & Bork, D. (2016a). Managing consistency in multi-view enterprise models: An approach based on semantic queries. In *Proceedings of the European Conference on Information Systems*.

Karagiannis, D. Buchmann, R. A., Burzynski, P., Reimer, U., & Walch, M. (2016b). Fundamental conceptual modeling languages in OMiLAB. In D. Karagiannis, H. C. Mayr, & J. Mylopoulos (Eds.), *Domain-specific conceptual modeling* (pp. 3-30). Berlin: Springer.

Karagiannis, D., Grossmann, W., & Höfferer, P. (2008). *Open model initiative: A feasibility study*. Retrieved from http://cms.dke.univie.ac.at/uploads/media/Open_Models_Feasibility_Study_SEPT_2008.pdf

Karagiannis, D., & Kühn, H. (2002). Metamodeling platforms. In K. Bauknecht, A. Min Tjoa, & G. Quirchmayr (Eds.), *Proceedings of the 3rd International Conference EC-Web* (LNCS 2455). Berlin: Springer.

Karagiannis, D., Mayr, H. C., & Mylopoulos, J. (2016c). *Domain-specific conceptual modelling*. Berlin: Springer.

Karagiannis, D., Buchmann, R. A., Walch, M. (2017). How can diagrammatic conceptual modelling supporting knowledge management? In *Proceedings of the European Conference on Information Systems* (pp. 1568-1583).

Karagiannis, D., & Buchmann, R. A. (2018). A proposal for deploying hybrid knowledge bases: The ADOxx-to-GraphDB interoperability case. In *Proceedings of the 51st Hawaii Conference on System Sciences* (pp. 4055-4064).

Koch, S., Strecker, S., & Frank, U. (2006). Conceptual modelling as a new entry in the bazaar: The open model approach. In *Proceedings of the IFIP International Conference on Open Source Systems*.

Langegger, A., & Wöß, W. (2009). XLWrap—querying and integrating arbitrary spreadsheets with SPARQL. In *Proceedings of the International Semantic Web Conference* (pp. 359-374).

Li, T., Grubb, A. M., & Horkoff, J. (2016). Understanding challenges and tradeoffs in iStar tool development. *ACM SIGSOFT Software Engineering Notes, 40*(6), 24-27.

Miron, E.-T., Muck, C., Karagiannis, D., & Götzinger, D. (2018). Transforming storyboards into diagrammatic models. In *Proceedings of the International Conference on Theory and Application of Diagrams*, (pp. 770-773).

Miron, E.-T., Muck, C., & Karagiannis (2019). Transforming haptic storyboards into diagrammatic models: The Scene2Model tool. In *Proceedings of the 52th Hawaii International Conference on System Sciences* (pp. 541-550).

Moody, D. (2009). The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, *35*(6), 756-779.

Mylopoulos, J. (1992). Conceptual modelling and telos. In P. Loucopoulos & R. Zicari (Eds.), *Conceptual modelling, databases, and CASE: An integrated view of information system development* (pp. 49-68). New York, NY: John Wiley & Sons.

Nuseibeh, B., Finkelstein, A., & Kramer, J. (1996). Method engineering for multi-perspective software development. *Information and Software Technology*, *38*(4), 267-274.

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., & Sinz, E. J. (2011). Memorandum on design-oriented information systems research. *European Journal of Information Systems, 20*(1), 7-10.

Popoola, S., Carver, J., & Gray, J. (2017). Modeling as a service: A survey of existing tools. In *Proceedings of the Model-Driven Engineering Languages and Systems Conference*.

Ralyté, J., Deneckère, R., & Rolland, C. (2003). Towards a generic model for situational method engineering. In J. Eder & M. Missiko (Eds.), *Proceedings of the 15th International Conference* on *Advanced Information Systems Engineering* (pp. 95-110). Berlin: Springer.

Recker, J. (2015). Research on conceptual modelling: Less known knowns and more unknown unknowns, please. In *Proceedings of the 11th Asia-Pacific Conference on Conceptual Modelling*.

Rolland, C. (2009). Method engineering: towards methods as services. *Software Process: Improvement and Practice*, *14*(3), 143-164.

Sabegh, M. A. J., & Recker, J. (2017a). Combined use of conceptual models in practice: An exploratory study. *Journal of Database Management*, *28*(2), 56-88.

Sabegh, M. A. J., Lukyanenko, R., Recker, J., Samuel, B., & Castellanos, A. (2017b). Conceptual modeling research in information systems: What we now know and what we still do not know. In *Proceedings of the 16th AIS SIGSAND Symposium.*

Sandkuhl, K., Fill, H. G., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A., Schwabe, G., Uludag, Ö., & Winter, R. (2018). From expert discipline to common practice: A vision and research agenda for extending the reach of enterprise modeling. *Business & Information Systems Engineering, 60*(1), 69-80.

Schuler, D., De Cindio, F., & De Liddo, A. (2015). Encouraging collective intelligence for the common good: How do we integrate the disparate pieces? In *Proceedings of the 7th International Conference on Communities and Technologies* (pp. 157-159).

Schwab, M., Karagiannis, D., & Bergmayr, A. (2010). i* on ADOxx: A case study. In *Proceedings of the 4th International i* Workshop* (pp. 92-97).

Stark, J., Braun, R., & Esswein, W. (2017). Systemizing colour for conceptual modeling. In J. M. Leimeister & W. Brenner (Eds.), *Proceedings of the 13th International Conference on Wirtschaftsinformatik* (pp. 256-270).

Utz, W., & Falcioni, D. (2018). Data assets for decision support in multi-stage production systems industrial business process management using ADOxx. In *Proceedings of the 16th International Conference on Industrial Informatics* (pp. 809-814).

van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2016). Open research in business and information systems engineering. *Business & Information Systems Engineering*, *58*(6), 375-379.

Visic, N., Fill, H-G., Buchmann, R. A., & Karagiannis, D. (2015). A domain-specific language for modeling method definition: From requirements to grammar. In *Proceedings of the 9th IEEE International Conference on Research Challenges in Information Science* (pp. 286-297).

von Briel, F., & Recker, J. (2017). Lessons from a failed implementation of an online open innovation community in an innovative organization. *MIS Quarterly Executive, 16*(1), 35-46.

Walch, M., & Karagiannis, D. (2017). Service-driven enrichment for KbR in the OMiLAB environment. In *Proceedings of ICServ (*LNCS 10371, pp. 164-177).

Walch, M., & Karagiannis, D. (2019). How to connect design thinking and cyber-physical systems: The s*IoT conceptual modelling approach. In *Proceedings of the 52nd Hawaii International Conference on System Sciences* (pp. 7242-7251).

Winter, R. (2010). *Business engineering navigator: Gestaltung und analyse von geschäftslösungen "Business-to-IT*". Berlin: Springer.

Yu, E. S.-K. (1995). *Modelling strategic relationships for process reengineering* (doctoral thesis). University of Toronto.

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal, 26*(3), 276-292.

About the Authors

Dominik Bork is currently working as a postdoctoral researcher at the Research Group Knowledge Engineering of the University of Vienna. Graduated with a diploma and a doctor rerum politicarum in information science from the University of Bamberg, Germany, he moved to Vienna in 2013. Dominik is heavily involved in the OMiLAB organization with a focus on international collaborations and metamodeling. He was involved in the conceptualization and the development of several conceptual modeling methods within the OMiLAB. Since 2018, he was elected domain expert of the Special interest Group on Modelling Business Information Systems of the German Informatics Society. His research works focus on the foundations of metamodeling, the specification of modeling methods, and the conceptualization of domain-specific modeling methods. He was visiting researcher at the University of Technology, Sydney, the Instituto Tecnologico Autonomo de Mexico, and the University of Pretoria.

Robert Andrei Buchmann received his doctoral degree in the field of E-commerce application models from Babeș-Bolyai University of Cluj Napoca, Romania, in 2005. Since then, he has been specializing in Semantic Technology and Enterprise Modeling, as enablers for Knowledge Management Systems or Enterprise Architecture Management. During 2012-2015 he occupied a postdoctoral research position at University of Vienna and came in contact with the OMiLAB innovation community, while specializing in the Agile Modeling Method Engineering framework and managing tasks for the ComVantage FP7 project. Currently, he occupies a Professor position at Babeș-Bolyai University where his team is working in OMiLAB-hosted projects, investigating opportunities of interplay between the paradigms of Semantic Web, Enterprise Modeling and Requirements Engineering.

Dimitris Karagiannis is full professor at the Faculty of Computer Science at the University of Vienna and head of the Research Group Knowledge Engineering. 2011 he was awarded an honorary professorship by the Babes-Bolayi University Cluj-Napoca in Romania. Moreover, he was a guest researcher at the Berkley University of California (USA), Kyoto University (Japan) and University of Technology Sydney (Australia). His main research areas are Business Process Management, Meta-Modelling, and Knowledge Management. He has published several books and scientific papers in journals and conferences on knowledge databases, expert systems, business process management, and knowledge management. In 1995, he established the Business Process Management Systems Approach (BPMS), which has been successfully implemented in several industrial and service companies. He is the founder and head of the supervisory board of the BOC Group (http://www.bocgroup.com) and a founding-member of the Open Models Initiative and has created 2012 the Open Models Laboratory (http://www.omilab.org).

Moonkun Lee is Professor in Division of Computer Science and Engineering in Chonbuk National University, Republic of Korea since 1996. He has a PhD degree in computer and information science. Worked at CCCC, USA, as Computer Scientist; Developed SRE (SW Re/reverse-engineering Environment) from 1992 to 1996; Applied to modernization of legacy OS and SW of NSWC in US Navy to Ada. His main research interests are SW round-trip engineering, distributed real-time systems, formal methods, ontology, behavior engineering, etc. Since 2015, as Head of OMiLAB KOREA, he developed focused SAVE and PRISM tools on the ADOxx Meta-Modelling Platform to model collective behavior of distributed mobile systems with δ-Calculus and Behavior Ontology. Currently focused on modelling IoT-Based Smart City and Factory with CPS (Cyber Physical Systems).

Elena-Teodora Miron is currently managing the OMiLAB NPO, the organization responsible for the OMiLAB ecosystem. Previously she was a project manager with focus on research and development projects in the fields of innovation and IT-supported management tools, both in academia and industry. She focuses on digital platforms and the platform economy as well as design thinking and open innovation methods. She has graduated in International Business Administration from the University of Vienna and is currently a PhD student in the same field.